

HTML Can Do That

HTML has been gobbling up swathes of what used to be JavaScript's remit. This page lists a bunch of dynamic functionality that we can now achieve with just HTML.

<dialog>

► Available since March 2022

Opened and closed entirely with `command` and `commandfor` HTML attributes that describe what “command” is being directed at which element, matched by `id`.

Open dialog

```
<button command="show-modal" commandfor="example-dialog">Open dialog</button>
<dialog id="example-dialog">
  <p>Look, Ma! No JavaScript!</p>
  <form method="dialog">
    <button type="submit">Close</button>
  </form>
</dialog>
```

popover

► Available since January 2025

Light dismiss, Esc to close, no managing z-index to wrangle it onto a top later. All managed with `popover` and `popovertarget` attributes in HTML.

Toggle popover

```
<button popovertarget="example-popover">Toggle popover</button>
<div id="example-popover" popover>
  <p>Same deal as the &lt;dialog&gt; above: no JS, just HTML attributes!</p>
</div>
```

Grouped <details>

- ▶ Available since September 2025

A shared name attribute turns a group of <details> into an exclusive accordion. Open one and the others close automatically. Magic!

- ▶ First
- ▶ Second

```
<details name="example-group">
  <summary>First</summary>
  <p>Open the seond one and watch this close on its own.</p>
</details>
<details name="example-group">
  <summary>Second</summary>
  <p>First one's hidden now.</p>
</details>
```

command & commandfor

- ▶ Available since December 2025

Two separate HTML buttons controlling *one* popover. No scripting.

```
<button command="show-popover" commandfor="example-command-popover">Open</button>
<button command="hide-popover" commandfor="example-command-popover">Close</button>
<div id="example-command-popover" popover>
  <p><code>show-popover</code> opens this and <code>hide-popover</code> closes
  it!</p>
</div>
```

Native input pickers

► Available since March 2017

Colour, range, and date pickers, built right into the browser.

Your mileage may vary with these elements; although, hand-written JS solutions tend to vary wildly, and I think we can expect form elements to receive more love over the coming years.

Colour

Range

Date 

```
<label>Colour <input type="color" value="#5f8aa6" autocomplete="off"></label>
<label>Range <input type="range" min="0" max="100" value="50" autocomplete="off">
</label>
<label>Date <input type="date" autocomplete="off"></label>
```

<datalist>

► Available since March 2019

Native autocomplete suggestions, no dropdown library required.

Favourite HTML element

```
<label>Favourite HTML element <input type="text" id="example-datalist-input"
list="example-datalist" autocomplete="off"></label>
<datalist id="example-datalist">
  <option value="a">
  <option value="abbr">
  <option value="address">
  <!-- ... -->
</datalist>
```

loading="lazy"

► Available since March 2022

This image defers loading until it's near the viewport. Not an [IntersectionObserver](#) in sight.



```

```

hidden until-found

► Available since December 2025

Navigating to the fragment link below reveals the hidden section.
The browser automatically removes `hidden="until-found"`.

[Jump to hidden content](#)

Yahaha! You found me!

```
<a href="#example-until-found">Jump to hidden content</a>
<div id="example-until-found" hidden="until-found">
  <p>Yahaha! You found me!</p>
</div>
```

Built by [Chris Burnell](#) for [HTML Day 2026](#) during the [Online Event](#) run by [Zachary Kai](#)
on Saturday, 8th August 2026.